Mirador: A Simple, Fast Search Interface for Remote Sensing Data

Christopher Lynnes, Richard Strub, Edward Seiler, Tilak Joshi and Peter MacHarrie

***Abstract*—A major challenge for remote sensing science researchers is searching and acquiring relevant data files for their research projects based on content, space and time constraints. Several structured query (SQ) and hierarchical navigation (HN) search interfaces have been developed to satisfy this requirement, yet the dominant search engines in the general domain are based on free-text search. The Goddard Earth Sciences Data and Information Services Center has developed a free-text search interface named Mirador that supports space-time queries, including a gazetteer and geophysical event gazetteer. In order to compensate for a slightly reduced search precision relative to SQ and HN techniques, Mirador uses several search optimizations to return results quickly. The quick response enables a more iterative search strategy than is available with many SQ and HN techniques.**

***Index Terms*—Database searching, information retrieval, remote sensing, search methods**

I. Introduction

EARTH Science researchers and other users of remote sensing data spend significant effort on two key functions: i) identifying data collections that support their research topic, and ii) reducing relevant inventories down to spatial and temporal ranges of interest. Many current Earth science search tools offer highly structured interfaces in order to ensure precise, non-zero results. The disadvantages of the structured approach lie in its complexity and resultant learning curve, as well as the time it takes to formulate and execute the search, thus discouraging iterative discovery. On the other hand, the success of the basic Google search interface shows that many users will forgo high search precision if the search process is fast enough to enable rapid iteration. However, most simple free-text search tools lack the rich ability to search on the key spatiotemporal metadata that allow researchers to zero in on the data they are looking for. At the Goddard Earth Sciences Data and Information Services Center (GES DISC), we have developed a search tool named Mirador (Spanish for "scenic overlook") that aims to provide the speed and simplicity of free-text search together with the capability to include spatiotemporal criteria. Mirador provides a search capability for the 257 datasets at the GES DISC, totaling 11 million data files in all. The chief goal is to provide a fast data discovery tool that is easy for researchers to learn and use, thereby reducing the time they spend hunting for data.

II. Mirador Search Paradigm

A. Comparison of Search Paradigms

Consider three prevalent search paradigms: hierarchical navigation (HN), structured query (SQ), and free-text search (FS). Hierarchical navigation is a progressive narrowing of search domains through menu selection, point and click, or even direct navigation of a file system or FTP site. In the early days of the World Wide Web (WWW), HN was commonly deployed, leading to a plethora of directories. HN is simple and intuitive, and it offers a high search precision, that is, a low percentage of unwanted results. Its chief disadvantage is scalability: as the diversity and number of potential results increases, the hierarchy deepens to keep each level manageable and navigable, lengthening the discovery process. Another disadvantage is the inability to execute simultaneous discovery in more than one branch of the hierarchy, preventing searches for more than one target at a time. This is less important when searching for documents, which can only be read one at a time, but is more burdensome for our problem domain, i.e., assembling a large, potentially diverse collection of individual data files.

Structured Query (SQ) is the process of searching a database for records whose attributes match desired ones entered by the user. In contrast to HN, SQ scales well to large numbers of discovery targets, i.e., millions. Also, a rich set of search attributes, together with Boolean operators, can support a variety of crosscutting searches. Expert users who know the metadata model can exploit this aspect of SQ to execute searches that are nearly as precise as HN. However, the richer the set of search attributes, the more a user must understand about the metadata model to achieve the desired results. Another disadvantage with SQ is that it is often difficult for the user to predict how many results will be returned for a given query. This is less of a problem if the user can query iteratively, adjusting constraints to achieve a manageable results set. However, we reach an unfortunate trade-off in very

Manuscript received January 25, 2008.

Christopher Lynnes, Richard Strub, Edward Seiler and Tilak Joshi are with the National Aeronautics and Space Administration, Goddard Space Flight Center, Greenbelt, MD 20771 USA (phone: 301-614-5185; fax: 301-614-5268; e-mail: Chris.Lynnes@nasa.gov).

Peter MacHarrie is now with Perot Systems, Suitland MD, 20746 USA (email: Peter.MacHarrie@noaa.gov).

large databases. Underconstraining the search by specifying only a few attributes often returns too many results and makes the catalog search take longer. Adding more attribute constraints makes the query formulation take longer and may then compound user frustration by returning no results.

Whereas numerical scalability is largely a function of database technology and hardware capability, scalability problems can arise with SQ in very heterogeneous search domains. As heterogeneity increases, the number of attributes common to all search targets decreases, i.e., the lowest common denominator. This leads either to a shrinking data model, with a resultant decrease in search precision, or an increasing number of exceptions to be accounted for.

Free-text search (FS) typically returns textual documents with occurrences of strings specified by the user. Like SQ, it scales well to large numbers of records. Also, FS is generally free of the heterogeneity scaling problems, since the simplicity of the data model for FS allows it to accommodate a huge diversity of records, indeed all the WWW. Another effect of the simple data model is that the user need know little or nothing about the database structure. As a result, the learning curve for FS is quite shallow. On the other hand, the disadvantage is the low precision of FS searches. This can be mitigated somewhat by the speed of both query formulation and search execution, allowing the user to experiment with different string combinations, usually implicitly linked via a Boolean AND operator, to obtain the desired results set. The shallow learning curve and scalability of FS in both numbers and diversity have made it the dominant search paradigm in the WWW at large, even to the point that one of the leading FS providers, Google, has entered the English language, among others, as a verb [1].

B. Search Paradigms in NASA's Earth Observing System

The Earth Observing System (EOS) is a collection of satellites, instruments and data systems that provide Earth science remote sensing data to the scientific research community. These data are typically stored as files (though some are in databases), which are in turn collected into groups, usually known as datasets or data collections. A common search task for Earth scientists is to find both the datasets of interest and the files within those datasets that correspond to the spatial area and time period representing the scientist's area of study. The relevant information describing the dataset includes such items as satellite name, instrument name, geophysical parameters and textual descriptions. In general, a dataset can be represented either as a document or records in a structured database. Files within a dataset typically share the same dataset-level metadata but differ from each other in spatial and/or temporal coverage. Space and time are not so easy to represent in documents in a way that can be easily searched, because they imply range-based comparisons. Spatiotemporal searches can be handled by SQ search engines simply by including spatial and temporal constraint attributes for the individual data files, together with the appropriate query support. HN can handle the temporal dimension through hierarchy (e.g. yearly directories, with daily subdirectories), but spatial hierarchies are more difficult to implement and harder still to integrate with a temporal hierarchy. The FS paradigm is even less amenable to the spatiotemporal constraints and rarely supports range searches.

Both the HN and SQ paradigms are well-represented in EOS. Many data centers offer HN based interfaces for locating and requesting data, ranging from simple anonymous FTP sites up to more elaborate WWW interfaces. The HN paradigm is particularly popular for datasets of global-coverage files, where only a temporal search (and therefore hierarchy) is needed. A hybrid paradigm (called the Web Hierarchical Ordering Mechanism) was implemented at the GES DISC to provide HN down to the daily level. At this point, a spatial query capability was grafted on to support spatial queries within that data day. The community welcomed its relative simplicity at the time, but some users complained about the clumsiness of navigating multiple trees for multiple datasets or time periods.

A much more elaborate SQ mechanism was developed by NASA in 1994. The EOSDIS Version 0 system provided a search across eight data centers distributed around the United States [2]. This EOSDIS Version 0 system supported searches on a variety of dataset attributes (e.g., instrument, satellite, geophysical parameter) as well as spatial temporal criteria. Over time, it evolved to the EOSDIS Data Gateway (EDG) [3] and now the Warehouse Inventory Search Tool (WIST), each using somewhat different technologies underneath but presenting consistent search experiences on the front end. Although the EOSDIS Version 0 architecture was a technical breakthrough at the time of introduction, some users have lamented the difficulty of query formulation and the slowness of search execution. Though some disadvantages are partially implementation-dependent (e.g., inefficient search response protocol), part of the difficulty is inherent in the SQ paradigm.

In the meantime, the explosion of free-text search in the general public has presented a tantalizing target for remote sensing data system developers. Two relatively early attempts to exploit the simplicity of the FS paradigm were the Mercury [4] and EOS-WEBSTER (http://eos-webster.sr.unh.edu) tools. Both of these are effectively SQ mechanisms, with a free-text search tab added on. This adds some ease of use to the search process, and query formulation can be done quickly using this free-text mechanism. Mercury searches usually end at the dataset layer, without providing query results for individual files matching space-time criteria. EOS-WEBSTER, on the other hand, does provide spatiotemporal querying of individual files among its 100,000-file data holdings. We hypothesized that a purely free-text search interface that provides a search experience similar to common public search tools would lower the learning curve even more. This was later reinforced by a vision statement within EOSDIS to move toward the use of common search tools for searching NASA remote sensing data

[5]. Mirador was developed to combine the shallow learning curve and quick query formulation of FS with the spatiotemporal query capabilities of SQ tools such as EDG and WIST.

III. Mirador Architecture

A. *Overview*

Mirador (http://mirador.gsfc.nasa.gov) combines a free-text dataset search with a relational database to store file-level information. The former is implemented using a Google appliance, searching dataset documents for user-specified keywords. This search is executed on a word index and is organized using Google's PageRank algorithm [6]. The algorithm ranks documents by attributes such as keyword density, keyword location, the number of incoming links to the page, and the number of outgoing links on the page. The relational database, implemented in the open-source database PostgreSQL, supports spatial and temporal queries for individual data items (usually files) within the datasets that pass the keyword screen. Thus, the metadata model used for searching is relatively simple: dataset-level metadata is encapsulated within the dataset documents, and the file-level metadata consists primarily of spatio-temporal information.

B. *Query Form Simplicity*

A key goal of Mirador was to simplify the query form for users. As a result, the front-end look-and-feel (Fig. 1) is designed to emulate the simplicity of free-text search interfaces. There are only five entry fields: Keywords, Location, two date/time fields for Time Span, and Event, all of which are of a simple text entry type. Only Keywords is required, so that a researcher can enter simply "SO2", without any time or spatial constraints, to search for data pertaining to sulfur dioxide.

For date/time values, the free-text entry field accepts any unambiguous time designation, parsing it to the most likely interpretation. Thus, a simple "2002" is a valid entry and becomes "2002-01-01 00:00:00" in the first (beginning) Time Span field, or "2002-12-31 23:59:59" in the second (ending) Time Span field.

The Location field supports not only entry of numerical latitude/longitude boxes or points, but also search-by-place-name using a gazetteer. Mirador's gazetteer stores geodetic points that are named (e.g., "New York") and classified (e.g., "County"). The toponymic information is based on the Geographic Names Data Base, containing official standard names approved by the United States Board on Geographic Names and maintained by the National Geospatial-Intelligence Agency. Mirador consolidates the name and (multiple and inconsistent) classification attributes of the publicly generated gazetteer entry into a single full-text attribute that is indexed using traditional (no page ranking) full-text indexing techniques. This combination of attributes and full-text indexing provides Mirador with a consistent FS feel and performance while utilizing a SQ database engine.

The Event field is similar, in that it supports searching by the names of geophysical events, such as named tropical storms. Searching on "Hurricane Katrina," for example, uses the space-time trajectory of Hurricane Katrina as space-time constraints in the search. Mirador's event gazetteer stores geodetic point and time attributes for geophysical events that are named (e.g., "Gabrielle") and classified (e.g., "Sub-Tropical Storm"). Tropical storm event information is from the Unisys Weather site (http://www.unisys.com/hurricane), and air quality event information is from the Environmental Protection Agency's AirNow site (http://www.epa.gov/airnow). By consolidating disparate event specific attributes (e.g., Air Pollution: Aerosol Type/Concentration and Named Storms: Category) into a single full-text attribute, Mirador provides a single interface to multiple geophysical event categories. With this approach, Mirador provides the user with a compact means of identifying remotely sensed observations that are specific to the researcher's area of interest.

C. *Search Optimization*

"In skating over thin ice, our safety is in our speed" – Ralph Waldo Emerson

The FS paradigm typically does not offer the high search precision that either HN or SQ can provide in the hands of an expert user. However, most FS tools compensate in two ways: relevance ranking algorithms and sheer speed. The first mechanism presents the results most likely to be relevant first. The second mechanism allows the user to experiment quickly to narrow, widen or re-orient searches. In Mirador, the relevance ranking begins with that returned by the Google appliance, which is then modified by excluding datasets from the initial results if they do not include files matching the user's space-time constraints.

The speed aspect is more complicated to achieve. It begins with quick query formulation resulting from the minimal form entry requirements. The subsequent keyword search of the database is also fast because the dataset documents number only in the hundreds. However, files number in the millions, and range searches against these can be time consuming, especially two-dimensional, cyclical searches like a spatial search. Yet retaining the speed of the FS paradigm is critical: user intolerance of slow browser applications can set in quickly, in as little as 12 seconds [7]. Quick response times are essential to enabling the iterative query refinement needed to compensate for the relatively low precision of FS queries.

In order to maintain the speed of a typical FS query, while still enabling spatial and temporal constraints, we have adopted a "lazy query" strategy for spatiotemporal searching, that is, to do the minimum query needed to present relevant results to the user. We separate the Mirador search into two parts (Fig. 2). The first, a Dataset Search, searches for those datasets whose metadata share a particular keyword, such as a name, instrument or physical parameter, and have files within the specified space and time constraints. If the user had to wait until the search engine retrieved all the file-level results matching the search constraints before presenting any results, it would be too time consuming to repeat the search with a modified constraint. Therefore, rather than return the individual files in a particular dataset, Mirador returns an estimated count based on the temporal and spatial characteristics of each particular dataset (Fig. 3). The hit estimate provides valuable feedback to help the user evaluate the results set for the space-time constraints.

Computing the hit estimate is simple for datasets whose files each cover the whole earth. For example, for the case of a daily global dataset, a time constraint of one year would return an estimate of 365 files. Results for orbital swaths or individual scenes are more complicated to estimate. Such data typically cover an area of the earth that is complicated to describe geometrically. For these estimates, Mirador uses an algorithm based on a spatial footprint table and a temporal tile table. The spatial footprint table stores geospatial information for each satellite footprint in a dataset's repeat cycle, where the footprint is the spatial area corresponding to an individual data file in a particular dataset, and the repeat cycle is the number of days after which a satellite-borne instrument retraces its ground track. The temporal tile table enumerates the days for which data files exist in the dataset. A spatial search for the user's bounding box is performed on the spatial footprint table to determine how many footprints intersect with the bounding box over the dataset's repeat cycle. The user's time constraint is compared against the temporal tile table to determine the number of distinct days for which data exist within that time span. The estimate is then obtained by:

$$Estimate = (M_r / T_r) \times T_d \qquad (1)$$

where M_r is the number of spatial matches in the repeat cycle, T_r is the number of days in the repeat cycle, and T_d is the number of days with data. The estimate is less accurate near the poles, particularly regarding the seasonal variation in spatial distribution for ultraviolet or visible radiance instruments. Nevertheless, it provides a rough order of magnitude estimate that is usually enough to judge the reasonableness of the results set for a given set of constraints.

The payoff of the hit estimator is the fast response time, usually about one or two seconds, even for loosely constrained queries with large results sets. This allows a researcher to skip entering space and time constraints in the search form, simply searching on one or more keywords until the proper set of datasets is identified, at which time, the query can be refined with space and time constraints. Thus, the search speed enables a crucial aspect of the interface simplicity.

The second part of the lazy query occurs when the user drills down to the file level for a particular dataset. File-level results are always presented in time-sorted fashion, allowing Mirador to execute a query with additional time constraints that fetch just enough results to fill one page. In combination with an index on time, the resulting query is quite fast, typically returning in one to four seconds. The time window of the query is shifted as the user navigates to each page of the results. This is analogous to a database cursor, but it does not require that the entire results set be identified before returning the first set of results and is independent of the cursor capabilities of any particular database management system.

The metadata model further accelerates the spatial search. In satellite remote sensing, it is common for several different files to be generated for one satellite footprint, each containing different geophysical parameters and belonging to a different dataset. This allows Mirador to use a single spatial metadata record to represent several files, which reduces the size of the database to be searched. At the dataset level, Mirador can even reuse the estimate for one dataset as the estimate for its sibling datasets without recomputing it. At the file level, the speed gain is obtained simply from the reduction in the database size, which is about a factor of two overall. Within this framework, satellite scenes bounded by quadrilaterals of arbitrary orientation are approximated with a bounding box whose sides are oriented along lines of latitude and longitude. More complicated geometries, such as satellite orbits, are represented via the union of up to 100 rectangles. As with the basic FS paradigm, these approximations sacrifice a small amount of accuracy, but again, compensate by being fast enough to support rapid iteration.

Overall then, the hope is that the combination of Mirador's short learning curve, minimal required fields and rapid search turnaround are enough to compensate for a potential loss of precision in searches. The idea is to allow the user to use a more iterative search strategy, manipulating keywords, time intervals and spatial boxes to obtain a manageable results set. This is particularly useful for users in the reconnaissance phase of a project, when they do not know exactly which datasets are most suited to their research problem.

IV. Conclusions

Ironically, a few users have commented that they "don't know what to type" into the Keywords field, indicating that the FS paradigm, while dominant in the world of search services, is not universally preferred. An automatic suggestion capability for the Keywords field will be implemented in the near future, but even this requires some initial typing. In response, we are beginning to add a capability to generate a hierarchical front-end, reusing the existing Mirador back-end as a search engine. This will allow users to switch between HN and FS paradigms as they like. The hierarchy will be generated using semantic web technology in the form of an ontology and a reasoner, in order to accommodate increasingly complex hierarchies of geophysical parameters and datasets. In addition, this technology will provide semantic mediation support to account for ambiguity, synonyms, and hypernyms in the FS part of the interface. Incorporation of semantic web technology will also enable machine-level interoperability with other semantically enabled systems, such as the Semantically-Enabled Science Data Integration project linking volcanic and atmospheric data [8].

Mirador currently provides search services only for data at the GES DISC. However, neither the search techniques nor the metadata model are specific to the GES DISC, although they are most applicable to remote sensing data. In fact, Mirador's dataset-level metadata are derived from entries in the Global Change Master Directory [9], and the metadata model is similar to that used by the Earth Observing System Metadata Clearinghouse (ECHO) [10]. Thus, one potential area for future work could be to provide Mirador search services over all EOS satellite data by ingesting metadata from ECHO.

ACKNOWLEDGMENT

We would like to thank Matthew Elkins and Louis Fenichel for their contributions to the Mirador software and database. We also appreciate helpful comments from Randy Barth, Bill Teng, and Bruce Vollmer.

REFERENCES

[1] F.C. Mish, ed., *Merriam-Webster Online*, [Online]. Available: http://www.m-w.com/dictionary/google.

[2] R. D. Price, M. D. King, J. T. Dalton, K. S. Pedelty, P. E. Ardanuy, and M. K. Hobish, "Earth science data for all: EOS and the EOS Data and Information System," *Photogrammetric Engineering & Remote Sensing*, vol. 60, no. 3, 1994, pp. 277-285.

[3] J. Yang and T. Johnson, "Metadata for Earth Observing Systems (EOS) Data Gateway," *Geoscience and Remote Sensing Symposium, 2000. Proc. IGARSS 2000. IEEE 2000 International*, vol.3, 2000, pp.1205-1207.

[4] R. Raskin, H. Burrows, H. Conover, J. Gallagher, G. Major and T. Rhyne, "Discovering and Accessing Data from the Federation of Earth Science Information Partners," *EOS*, vol. 83, no. 47, Electronic Suppl., 2002. Available: http://www.agu.org/eos_elec/020137e.html

[5] M. Esfandiari, H. Ramapriyan, J. Behnke, and E. Sofinowski, "Evolving a ten year old data system," *2nd IEEE International Conference on Space Mission Challenges for Information Technology*, Pasadena, CA, 2006, pp. 243-250.

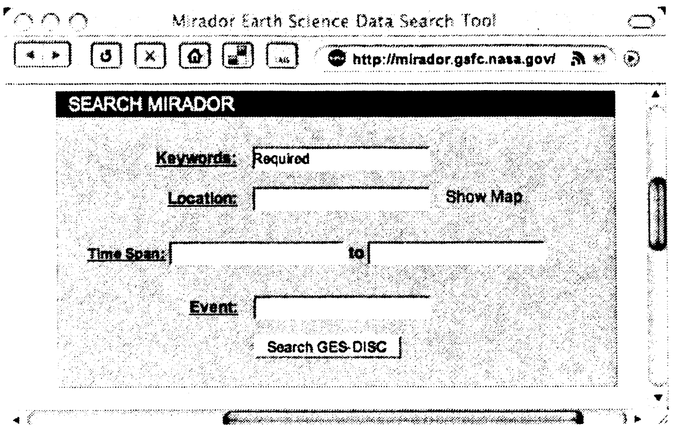
[6] S. Brin and L. Page. "The anatomy of a large-scale hypertextual web search engine," in *Proc. of the 7th International World Wide Web Conference*, Brisbane, 1998, pp 107–117.

[7] J. A. Hoxmeier and C. DiCesare, "System response time and user satisfaction: An experimental study of browser-based applications," in *Proc. of the Association of Information Systems Conference*, Long Beach, CA, 2000.

[8] P. Fox, D. L. McGuinness, R. Raskin, and K. Sinha, "A Volcano Erupts: Semantically Mediated Integration of Heterogeneous Volcanic and Atmospheric Data," in *Proc. of the First Workshop on Cyberinfrastructure: Information Management in eScience*, Lisbon 2007.

[9] L. M. Olsen and G. R. Major, "Global Change Master Directory Enhances Search for Earth Science Data," *Trans. EOS, Am. Geophysical Union*, Available: http://www.agu.org/eos_elec/95127e.html.

[10] R. Pfister, R. Ullman and K. Wi*chmann*. "ECHO Responds to NASA's Earth Science User Community," in *Proc. HCI International 2001: 9th International Conference on Human-Computer Interaction*, New Orleans, LA, 2001.


Mirador Earth Science Data Search Tool

http://mirador.gsfc.nasa.gov/

SEARCH MIRADOR

Keywords: Required

Location: Show Map

Time Span: to

Event:

Search GES-DISC

Fig. 1. Mirador search form. All fields are free-text. Location can consist of place names or latitudes and longitudes. Time spans can be specified in any non-ambiguous format. Event represents geophysical event names, such as named tropical storms. Only the Keywords field is required.

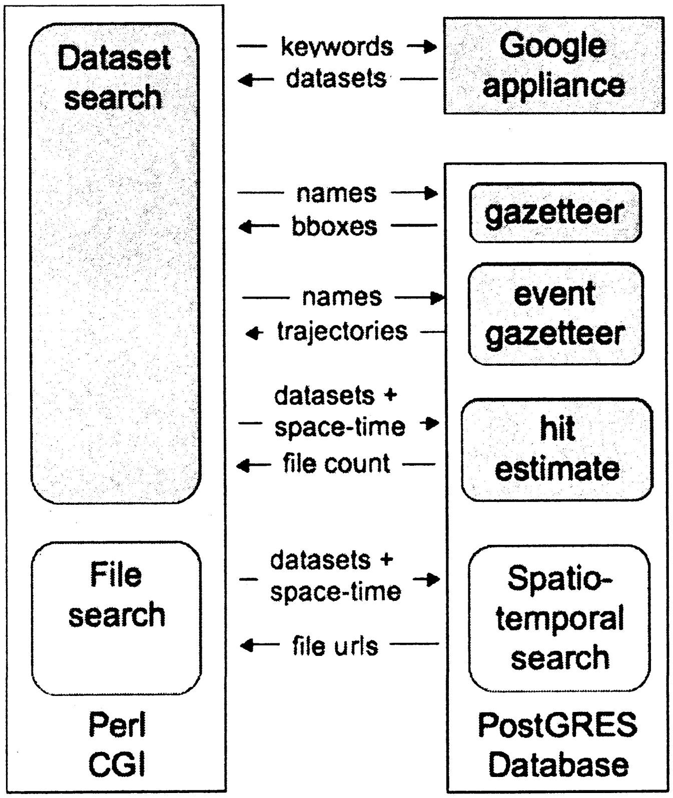
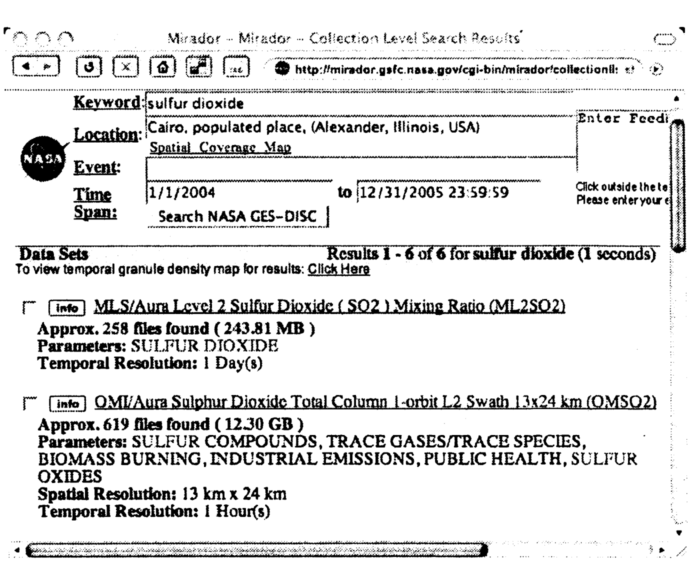


Fig. 2. Architecture for Mirador. Boxes represent underlying technologies; rounded rectangles are functional elements.


Mirador - Mirador - Collection Level Search Results

http://mirador.gsfc.nasa.gov/cgi-bin/mirador/collectionli:

Keyword: sulfur dioxide

Location: Cairo, populated place, (Alexander, Illinois, USA)

Spatial Coverage Map

Event:

Time Span: 1/1/2004 to 12/31/2005 23:59:59

Search NASA GES-DISC

Enter Feedi

Click outside the te
Please enter your e

Data Sets Results 1 - 6 of 6 for sulfur dioxide (1 seconds)

To view temporal granule density map for results: Click Here

□ info MLS/Aura Level 2 Sulfur Dioxide (SO2) Mixing Ratio (ML2SO2)

Approx. 258 files found (243.81 MB)
Parameters: SULFUR DIOXIDE
Temporal Resolution: 1 Day(s)

□ info OMI/Aura Sulphur Dioxide Total Column 1-orbit L2 Swath 13x24 km (OMSO2)

Approx. 619 files found (12.30 GB)
Parameters: SULFUR COMPOUNDS, TRACE GASES/TRACE SPECIES, BIOMASS BURNING, INDUSTRIAL EMISSIONS, PUBLIC HEALTH, SULFUR OXIDES
Spatial Resolution: 13 km x 24 km
Temporal Resolution: 1 Hour(s)

Fig. 3. Close-up of Dataset Results page for search on "sulfur dioxide", "Cairo, Illinois and time span 2004 through 2005.